Xiaoqi Lei

Determining geographic origin of social media users with Bayesian Analysis of common syntactical and spelling errors when using foreign languages

By

Xiaoqi Lei

Supervisor:
Jaran Nilsen
Ole-Christoffer Granmo

Master Thesis in
Information and Communication Technology

Faculty of Engineering and Science
University of Agder

Grimstad, 25st May 2011

Abstract

As the growing influence and importance of social media, the need of categorizing authors of overt text information from social media by their geographic origin background is becoming more urgent than ever before. To achieve the goal, some method been developed, for instance, classifying by authors' language, timezone, or by geographic terms used in the text.

This thesis explored a unique classifier to determine the social media users' geographic background: Native Language Classifier, which classifies authors' native language from the text they have written in English.

The Native Language Classifier set up a training set consisting of English corpus in size of 6 million words of 800 authors from 4 different language background: Chinese, Russian, Spanish and French. And through testing 200 users (50 users from each language group) the classifier made an overall accuracy of 75% by combining result from n-gram algorithms in word level, n-gram algorithms in character level, and spell checking algorithm, to classify those authors into groups of correct language background. It would be valuable for both social media analyzers, and text classifying researchers.

More than the classifying result, some interesting observations are made from the test as well. They disclosed some rules behind the languages. Therefore the method developed by this thesis would also possibly become a useful tool to help researchers analyzing the feature of the languages.

Preface

This report is submitted in fulfillment of the requirements of the degree Master of Science at University of Agder (UiA), Faculty of Engineering and Science, Grimstad, Norway. The project is supported by Integrasco A/S, which has provided corpus material and supporting frameworks which has been used to carry out various parts of the study. Supervisor on the project has been Ole-Christoffer Granmo at UiA and Jaran Nilsen at Integrasco A/S.

I would like to thank Ole-Christoffer Granmo for his excellent supervision and guidance. More than that, the course "Machine Learning" taught by him at UiA led me to research in this field, and inspired my interests to select the topic of this report. Also great thanks to Jaran Nilsen, for his great support throughout the project period. His previous study and experience gave me a lot of inspiration, his input and feedback is also invaluable for this research.

Grimstad, May 23rd, 2011

Xiaoqi Lei

Contents

1 Introduction

1.1 Background

The topic of determining geographic origin of social media users is raised as the growing of social media in recent years.

"Social media are media for social interaction, using highly accessible and scalable communication techniques. Social media is the use of web-based and mobile technologies to turn communication into interactive dialogue. Andreas Kaplan and Michael Haenlein also define social media as "a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, which allows the creation and exchange of user-generated content." Businesses also refer to social media as consumer-generated media (CGM). "[1]

From a statistic published by D. Steven White on his blog "All Things Marketing" about the numbers of users of major social media sites[2], we can see the growing speed of social media.

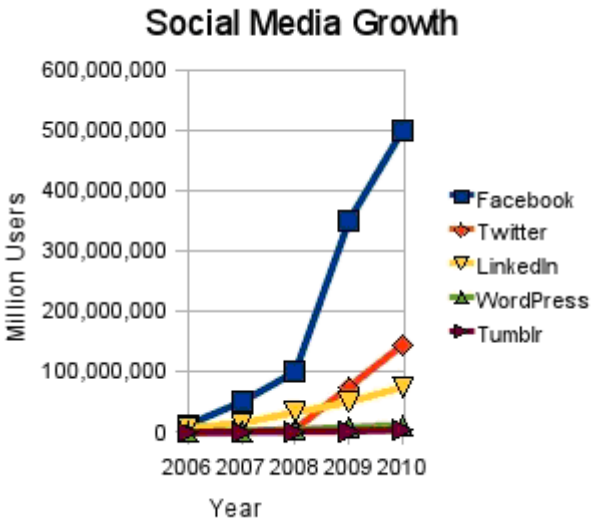

Chart 1-1 Social Media Growth

And from a summary published by Laura Frangi on the blog "We are social" about the UK's media consumption habits[3] in Chart 1-2, we can really feel the impact have made by social media. More than 60% of people from 15 years old to 34 years old access to social network on the internet in UK.

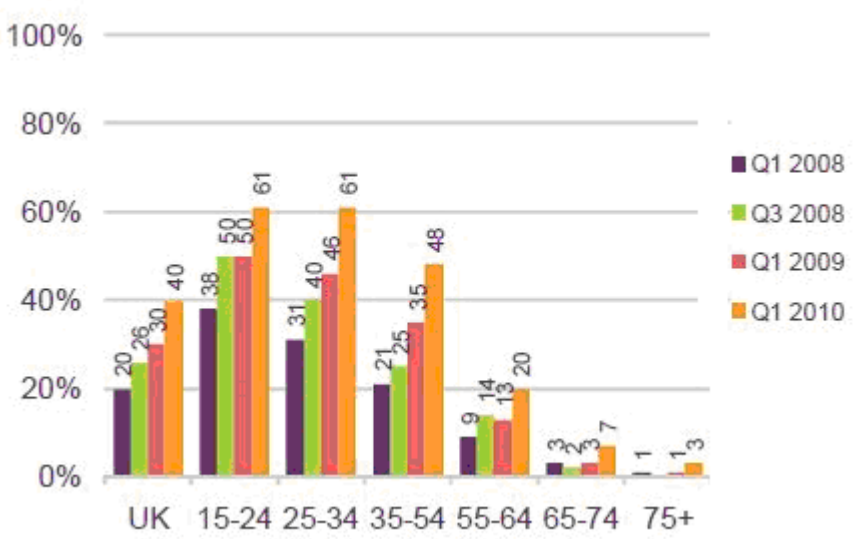

Chart 1-3 Proportion of adults who access social Networking sites on the internet at home (UK)

As the evidence that social media has become the most important platform for people (especially younger people) to create and share information, it has also become an irreplaceable field to be studied in on varieties of topics for either academic or business purposes as a matter of course. And to effectively utilize the information obtained from social media, there is a need to classify social media users depending on the requirement of the research.

The classification standards of social media users are diverse: gender, age, profession, interest, race, or location etc. The geographic location is frequently used as classification standard when a research is depending on regions.

Some research has been done on the topic of identifying the locations of social media users.

Jaran Nilsen has developed an algorithm applied 3 indicators to identify locations of social media users: Language Indicator, which is base on identifying the language of the text; Time-zone Indicator, which could indicate which timezone are the authors from; and Geoterm Indicator, which suggests the authors' location by the geographic terms shown in the text. Referring to the test of this research, Geoterm Indicator doesn't provide good accuracy and is low on efficiency, so it's ignored in the combined test. The Language Indicator and Time-zone Indicator performed very well.[4]

Ole-Alexander Moy has developed a possible method to apply Geoterm Indicator. 3 indicators are selected in the surrounding text of a possible geographic term: Five-pre-terms Indicator, First-pre-term Indicator, and Five-post-terms Indicator, and applied Naïve Bayes Classifier to classify text according those indicators. It gives very good result.[5]

Prior to this study, some research are also made on the locations identifying of social media users before doing research on this thesis. It has improved Ole-Alexander Moy's algorithm to suggest location based on geographic terms by theoretically consummate it according to the Bayesian Theory.

But there are limitation of those indicators mentioned above: Language Indicator won't be able to work when a group of authors from different countries but all speak English; The Timezone Indicators will meet problem when author's schedule is different from others; And the GeoTerm Indicator only works when authors referred the name of their location.

This thesis will develop a classifier which can be called "Native-language Classifier" to identify author's location when the methods provided above don't work. Or it could also work together with those methods to make the result even better.

1.2 Hypothesis

The approach is based on the assumption that people will be affected by their native language when they speak foreign languages more or less. Thus, there will be common features existed between those people who come from the same place and share the same native languages when they are speaking a specific foreign language, for example, English. The features could be the habits to use some words, or use certain sentence patterns or even tendency to make similar mistakes. And we are able to identify those features by making statistic, and apply Naive Bayes Classifier to classify target text according those features.

1.3 Problem statement

Words, phrases, sentences, grammar, speech, writing, and alternative symbols, are all important elements of a natural language[6]. Each language has its distinct feature.

The problem will be discussed in the thesis is: how can we identify those common features from people have same background of native language when they are speaking English, and apply them properly to classify social-media authors of English text into groups of different native language speakers, which would indicate the possible locations where they are from?

1.4 Report outline

The rest of this thesis is organized as follows; Chapter 2 contained the theoretical background information introduced what concepts and technologies I have adopt to solve the problem. Chapter 3 introduced the solution in order of the steps of the key procedure; Chapter 4 showed the test result and made some discussion around it. Chapter 5 made a conclusion and summarized the potential field to be developed in advance.

2 Theoretical background

2.1 Pattern Classification

The task of pattern classification is to place different discovered patterns into groups. Humans are doing pattern classification almost all the time. For example, people may classify things according to their type, size, shape, colour, or so. And people classify sound, classify some abstract concepts. Basically everything in the world will be classified naturally by human according to variety of patterns. But for machine, it might be much more difficult.

The figure below shows a usual approach of pattern classification:

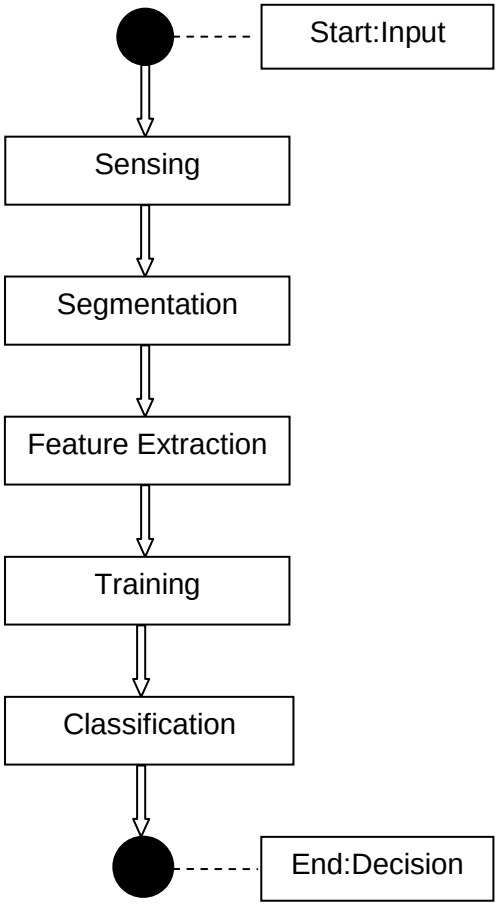

Figure 2-1 The common steps of pattern classification

Though it's not the only approach of pattern classification, but there are some core elements shared by all those approaches: Feature Extraction, Training, and Classification.

Feature extraction for an object can be done in different ways depending upon how to measure features for the object at hand. In the case which is going to be presented in the report, we will use the text surrounding the possible geographic term as the feature to be extracted. Then we can use those features

Then these features can be used to classify the input text. The decision for which group the input pattern belongs to can be made based on many different decision theories. For the case in this research, Bayesian Decision Theory is adopted.[7]

2.2 Naive Bayes Classifier

"Naive Bayes classifier is a simple probabilistic classifier based on Bayes' theorem with naive independent assumptions. It assumes the presence of a particular feature of a class is independent to the presence of any other feature. We consider a probability model for a classifier as a conditional model:

$$p(C|F_1,\dots,F_n)$$

"Over a dependent class variable C with a small number of outcomes or classes, conditional on several feature variables F_1 through F_n. The problem is that if the number of features n is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.
Using Bayes' theorem, we write

$$p(C|F_1,\dots,F_n) = \frac{p(C)\ p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}.$$

"In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$p(C, F_1,\dots,F_n)$$

"which can be rewritten as follows, using repeated applications of the definition of conditional probability:

$$\begin{aligned}
&p(C, F_1,\dots,F_n)\\
&= p(C)\ p(F_1,\dots,F_n|C)\\
&= p(C)\ p(F_1|C)\ p(F_2,\dots,F_n|C,F_1)\\
&= p(C)\ p(F_1|C)\ p(F_2|C,F_1)\ p(F_3,\dots,F_n|C,F_1,F_2)\\
&= p(C)\ p(F_1|C)\ p(F_2|C,F_1)\ p(F_3|C,F_1,F_2)\ p(F_4,\dots,F_n|C,F_1,F_2,F_3)\\
&= p(C)\ p(F_1|C)\ p(F_2|C,F_1)\ p(F_3|C,F_1,F_2)\ \dots p(F_n|C,F_1,F_2,F_3,\dots,F_{n-1}).
\end{aligned}$$

"Now the 'naive' conditional independence assumptions come into play: assume that each feature F_i is conditionally independent of every other feature F_j for $j\neq i$. This means that

$$p(F_i|C,F_j) = p(F_i|C)$$

"for $i\neq j$, and so the joint model can be expressed as

$$\begin{aligned}
&p(C, F_1,\dots,F_n) = p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots\\
&= p(C)\prod_{i=1}^{n} p(F_i|C).
\end{aligned}$$

"This means that under the above independence assumptions, the conditional distribution over the class variable C can be expressed like this:

$$p(C|F_1,\dots,F_n) = \frac{1}{Z}p(C)\prod_{i=1}^{n} p(F_i|C)$$

"where Z (the evidence) is a scaling factor dependent only on F_1,……F_n, i.e., a constant if the values of the feature variables are known.
Models of this form are much more manageable, since they factor into a so-called class prior p(C) and independent probability distributions $p(F_i|C)$. If there are k classes and if a model for each $p(F_i|C=c)$ can be expressed in terms of r parameters, then the corresponding naive Bayes model has (k − 1) + n r k parameters. In practice, often k = 2 (binary classification) and r = 1 (Bernoulli variables as features) are common, and so the total number of parameters of the naive Bayes model is 2n + 1, where n is the number of binary features used for classification and prediction.

"The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule. The corresponding classifier is the function classify defined as follows:

$$\text{classify}(f_1, \ldots, f_n) = \underset{c}{\text{argmax}}\, p(C = c) \prod_{i=1}^{n} p(F_i = f_i | C = c).$$

" [8]

2.3 N-gram analysis

N-gram is a subsequence in which each unit consists of n items from a given sequence. The items to form the sequence depends on what object is to be analyzed. One of the typical uses is text categorization. For text categorization, the sequence could be either in word level or character level.

For example, to analyze the text *"All of the folks that I love are there"*, the set of 2-grams (bigrams) in word level would contain: *"All of", "of the", "the folks", "folks that", "that I", "I love", "love are", and "are there"*. And the set of 3-grams (trigrams) in character level would contain: *"All", "ll ", "l o", " of", "of ", "f t", " th", "the", "he "*, or so.

The n-gram has many advantages. First, it doesn't need any prior knowledge on a language to apply it, no grammar, and no dictionary. Second, it's language-neutral, which means it could be applied on any languages. Third, the most important one, it provides good garble tolerance performance. It's especially important when we want to apply it on the social media. Because we can imagine there will be tons of mistakes and non-standard usages of languag.e in the text from social media. N-gram is just the right tool to analysis it.

But the drawbacks of n-gram are obvious as well. First, it lacks any explicit representation of long range dependency. Thus n-gram will be not able to do grammar analysis in a level of sentences. Second, when the n of n-gram grows, the requirement of memory space grows in geometric progression.[9]

2.4 Spell Check

The purpose of spell check is to find out the wrongly spelt words. In this thesis, we selected Jazzy API –a GPL/LGPLed Java-based spell checker API based on the Aspell algorithm.

The Aspell algorithm is an algorithm combined edit distance algorithm and phonetic matching algorithm, which are 2 main direction of ideas of spell check. Now we'll explain the 2.

2.4.1 Edit Distance

"Edit distance which is also known as 'Levenshtein distance' is a measure of the similarity between two strings, the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t. The greater the edit distance, the more different the strings are. In our case, the source string is the input, and the target string is one of the entries in the dictionary." [10]

To implement edit distance algorithm in spell check, the typical method is: First, build a dictionary of all the words correct spelt. Second, compare the wrongly spelt word with the words in the dictionary one by one, and return with the distance of each word. Last, suggest those words have the nearest distance to the target wrongly spelt word.

To calculate the distance between 2 strings, the usual method is resulting matrix. For example, if we want to compare "diverse" and "device", we can make a resulting matrix like the table "Table 2-1Edit Distance Algorithm" below:

		d	i	v	e	r	s	e
	0	1	2	3	4	5	6	7
d	1	0	1	2	3	4	5	6
e	2	1	1	2	3	4	5	6
v	3	2	2	1	2	3	4	5
i	4	3	3	2	2	3	4	5
c	5	4	4	3	3	3	4	5
e	6	5	5	4	4	4	4	4

Table 2-1Edit Distance Algorithm

We start with filling the cell in the left-up corner with 0. Then we decide its neighbour cells' value by whether it takes a change to make the substrings in the first column and the first equal to each other according the position of the current cell, following the order from the left to the right and from the top to the bottom. If it needs a change, increase the value by one, else keep it the same.

When the entire matrix if finished, the number in the right-bottom corner will be the overall number of required changes. The path marked by the arrow in the figure shows the different steps to transform one string to the other. Obviously, the path is not unique for given pair of Strings.

The edit distance is a widely used algorithm for string comparing and spell check. It's easy to implement. But it's not perfect. A lot of words are quite similar to each other from the view of edit distance algorithm, but few people will really mix them together. For instance: "right", "fight", "night", and "tight". They have totally different meaning, and no one would seem to wrongly spell one of them to another, though the distance from one of them to another is so close. So when we apply the edit distance algorithm on spell checking, it doesn't always make good guess on a wrongly spelt word.

2.4.2 Phonetic matching

"The rich variation in the spelling of names, in particular, has led to some interesting spell checking algorithms. A typical one is the phonetic matching algorithm, which aims to solve the problem of 'which names match those that sound like x.' This algorithm type is quite common in search databases and other reference applications." [11] One of the typical phonetic algorithms is Soundex.

"The Soundex code for a name consists of a letter followed by three numerical digits: the letter is the first letter of the name, and the digits encode the remaining consonants. Similar sounding consonants share the same digit so, for example, the labial consonants B, F, P, and V are each encoded as the number 1. Vowels can affect the coding, but are not coded themselves except as the first letter. However if "h" or "w" separate two consonants that have the same soundex code, the consonant to the right of the vowel is not coded.

"The correct value can be found as follows:

1. The first letter of the name is the letter of the Soundex code, and is not coded to a number.
2. Replace consonants with digits as follows (after the first letter):
    * b, f, p, v => 1
    * c, g, j, k, q, s, x, z => 2
    * d, t => 3
    * l => 4
    * m, n => 5
    * r => 6
    * h, w are not coded
3. Two adjacent letters with the same number are coded as a single number. Letters with the same number separated by an h or w are also coded as a single number.
4. Continue until you have one letter and three numbers. If you run out of letters, fill in 0s until there are three numbers."[12]

The Soundex algorithm's major drawback when applied on spell check is that it is suggest all the words which sounds alike instead of sounds precisely distinct. In other word, it gives too much rough matches.

3 Solution

3.1 Training set building

A good training set is critical for the performance of a Bayesian text classifier. Consequently to build a good training set and refine it to better and better is a very important job from the beginning to the end of the programming task.

What is a good training set? For a Bayesian text classifier, a training set considered to be good means the corpus in the set has strong representativeness for the group it belongs. According to this requirement, some preferences of selecting corpus for training sets are made. We'll discuss them one by one.

3.1.1 English proficiency of authors

If we want to make a training set of English speaking authors whose native language is French, those author who speaks perfect English would not be preferred. Because it would be too difficult to find any proof or sign of his/her native language – French.

Instead, a new English learner is more probable to have his native language exposed. Misspellings, grammar mistakes, or even some accustomed expressions, all above, might have a relationship from his/her native language, more or less.

In the other hand, the low English proficiency authors are not the only group to be added into the training set. Actually a good training set should cover authors of different level of English proficiency, except for perfect English speakers. Because we should aware that a better proficient English speaker may have chance to make mistakes which lower proficient English speakers have less chance to make.

3.1.2 Cultural environment of authors

Those authors who lived in a foreign country with different cultural environment are not preferred. Because one live in a different culture might be affected when speaking or writing in English. For instance, a Chinese lived in Norway for many years, he/she might speak English close to a Norwegian more or less. A more extreme example is, an American born Chinese, maybe he/she can speak perfect English, but not able to speak any Chinese in the mean time. In this example, the native language has barely anything influence to his/her English speaking/writing.

In short, the people born, lived and spent most of time in his/her motherland is preferred.

3.1.3 Quality of single post

The corpus for this research is collected from the social media. Which means the quality of each post is variable. There are some factors might affect the quality of a posts which need to be aware.

First, quoted contents. The quoted contents in a reply are sometimes redundant content, which may cause a bias of the training set, because the quoted contents can be quoted for more than once. More than that, the quoted contents could be pure noises, for we wouldn't know any background of the author who is quoted. Therefore, the quoted contents in a post must be removed before we use the post.

Second, the length of single post. Too short or too long posts are not preferred. Too short post usually doesn't carry out much information, and brings noises to the training sets. Too long post is very possible to be a repaste of news or someone else's passage.

3.1.4 The size of training set

As the corpus material has a finite size, and uneven quality, how to decide the size of the training set is an interesting problem. Finite in size and uneven quality imply it's not quite possible to make an ideal training set which have perfect representativeness and high efficient which implies small size in the meantime.

In the practice, a relatively larger size of training set might implies a better coverage, but may also brings more noise and lower efficient; a smaller size of training set might implies a less coverage, but it can also have better precision if constructed well and the efficiency should be better than a larger set. Anyway, it required a balance on the size between representativeness and efficient, and it needs tests to find out a proper size.

3.2 N-gram algorithm

The purpose implementing n-gram algorithm is to expose the relations between words to words, characters to characters. Though as it is said in Chapter 2.3, the n-gram algorithms are almost blind to find the long range dependency in sentences, it is still quite sensitive in detecting the short range combination. The most interesting thing is that when we apply the n-gram algorithm in different depth or based on different unit, we are actually examining the text in a different aspect.

3.2.1 N-gram in word level

Applying n-gram in word level is a view focusing on words, and the combinations of words.

Reviewing the text in the unigram (1-gram) in word level is like making a vocabulary with frequency of each word. It indicates the preference when people selecting words to express. And it also reflects the words of high frequency of misspelling for authors of each native language.

Reviewing the text in the n-gram where n is larger than 1 in word level, focuses more on the way people connecting words, for instance, using phrases or even short sentences (some are colloquial sentences). As the n growth larger, the number of combinations increases in geometric progression, and also because we don't really need that much, we limit the n to be equal or less than 4 in this research.

3.2.2 N-gram in character level

Applying n-gram in character level will help us find the habit people spelling words, and highly reappearing spelling mistakes. Or we can conclude it as "spelling style".

For instance, in British English, words are spelt as: "litre", "metre", "centre", "theatre", and so on. In the meantime, those words are spelt to be:"liter", "meter", "center", and "theater" in American English. It's easy to find out that there is a high rate of appearing "-tre" in British English, and "-ter" in American English. We can observe the same of "our" in British English for "colour", "armour", "favour", and "or" in American English for "color", "armor", "favor" as well. The n-gram algorithm in character level is quite sensitive to detect those little pieces in spelling.

And taking account to the misspelling caused by language background, for example, the English word "history" is spelt as "histoire" in French, the English word "authority" is spelt as "autorité" in French. They are all similar words, and easy to be misspelled by a person have French background, but n-gram algorithm would be able to detect them.

Considered the same reason explained in Chapter 3.2.1, the depth of n of "n-gram" is limited to be equal or less than 4. And the unigram in character level would be just a set made of 26 letters with frequency of each letter, which doesn't make sense, so it is not selected as one of the algorithms in the research.

3.2.3 Classifier building

According to Chapter 3.2.1 and 3.2.2, we selected n-gram algorithms in word level (n=1,2,3,4), and n-gram algorithms in character level (n=2,3,4) to form the classifier.

3.3 Spell check algorithm

By making spell check on a given text, we hope to find out all the spelling mistakes. It's possibly indicating what the highly misspelled words are for each group of authors of certain native language.

As it is introduced in Chapter 2.4, Jazzy is an open-sourced spell checking API using a combined algorithm of both phonetic matching and edit distance algorithm. The jazzy API returns the distance value of the 2 parameter of input strings. In our research, we compare each word in the given text with the words in the dictionary, and select the word have the nearest distance.

After that, we use the number occurrences of each word to form the Naive Bayes Classifier.

3.4 Result combination

Provided we have made 8 algorithms in total (4 n-gram algorithms in word level, 3 n-gram algorithms in character level, 1 spell check algorithm), we need to utilize the result from the 8 algorithms to make a final result.

Here we can assume that a set of some algorithms might work better for certain languages, and a set of some other algorithms might work better for other languages. Thus we want to assign a weighted factor to each algorithm for each language. Then make a vote with the value of multiplication of the assigned weight and the result from each algorithm to determine which set of native language the testing author belongs.

Specifically, we developed a formula from the experience of our testing. We index the 4 language backgrounds as L_i (i=1~4), and index the 8 algorithms as A_j (j=1~8). And provided applying algorithm A_j, accuracy number of language L_i is N_{ji}, we assign the weighted value for Language L_i and A_j as W_{ji}:

$$Wji = \left(\frac{8 * Nji}{\sum_{j=1}^{8} Nji} \right)^5$$

This formula is developed from the experience of our testing to make the algorithm, and performed well in most of the condition.

4 Test Result and Discussion

4.1 Validation and Testing

To find out the best size of training sets, we have set up some tests of different size of training set. The first set contains about 0.5 million words for each group of authors shared the same native language which are Chinese, Russian, Spanish, and French. We name it as A set. The second set contains about 1.5 million words for each group, and we name it as B set. The second set contains about 5 million words for each group. We name it as C set. Here we should notice that because of the limited resource of corpus, the threshold of selecting authors into training set C is lowered to collect enough text. For instance, those people who speak a certain language but not clearly known to be the citizen of a country speaking that language are accepted. But according to Chapter 3.1, it might brought noises into the training set, and affects the performance.

Base on each training set, we selected 200 users in total to form the test set (50 for each language). And we've made a confusion matrix for each algorithm on each set. And for each algorithm they are compared in the Chapter 4.2:

4.2 Unigram (1-gram) in word level

Set A (0.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	24	6	8	12
	Russian	5	11	10	24
	Spanish	9	0	21	20
	French	5	4	12	29
Overall accuracy:		**43%**			

Table 4-1 Unigram in word level for set A

Set B (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	30	8	8	3
	Russian	12	23	10	5
	Spanish	6	3	38	2
	French	11	8	15	16
Overall accuracy:		**54%**			

Table 4-2 Unigram in word level for set B

Set C (5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	27	10	11	2
	Russian	5	25	13	7
	Spanish	5	10	32	2
	French	7	4	20	19
Overall accuracy:		**52%**			

Table 4-3 Unigram in word level for set C

We can see from the comparison that for set A, it doesn't work well with Russian and French. We have 50 French authors to be tested, but more than 80 authors are identified to be French. Also we have 50 Russian to be tested, but just 21 authors are identified to be Russian overall.

For set B and C, the results are better. Both have better accuracy than set A.

4.3 Bigram (2-gram) in word level

Set A (0.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	28	6	6	10
	Russian	5	15	15	12
	Spanish	5	2	28	15
	French	5	4	17	24
Overall accuracy:		**48%**			

Table 4-4 Bigram in word level for set A

Set B (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	33	4	8	4
	Russian	5	30	10	4
	Spanish	1	1	39	8
	French	6	10	15	19
Overall accuracy:		**61%**			

Table 4-5 Bigram in word level for set B

Set C (5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	27	8	12	3
	Russian	6	29	10	4
	Spanish	3	10	31	5
	French	4	6	19	21
Overall accuracy:		**54%**			

Table 4-6 Bigram for set in word level C

All of the 3 sets work better than unigram algorithm in word level. But set A still perform worse than the other 2 sets.

It is obvious to be noticed that the bigram algorithm in word level works very well for both Chinese and Spanish in any one of the 3 sets. And the results on Russian and French are not bad either.

4.4 Trigram (3-gram) in word level

Set A (0.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	34	6	7	3
	Russian	7	20	5	12
	Spanish	8	6	18	17
	French	5	5	7	32
Overall accuracy:		**52%**			

Table 4-7 Trigram in word level for set A

Set B (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	31	4	7	7
	Russian	3	31	7	8
	Spanish	5	2	31	11
	French	6	12	19	13
Overall accuracy:		**53%**			

Table 4-8 Trigram in word level for set B

Set C (5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	20	13	9	7
	Russian	5	28	7	6
	Spanish	3	13	27	6
	French	3	8	17	21
Overall accuracy:		**48%**			

Table 4-9 Trigram in word level for set C

The trigram algorithm in word level performs not such as good as the bigram algorithm, but either not bad for any of the languages.

4.5 4-gram in word level

Set A (0.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	17	8	5	10
	Russian	3	18	0	6
	Spanish	7	7	10	22
	French	5	4	4	29
Overall accuracy:		**37%**			

Table 4-10 4-gram in word level for set A

Set B (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	15	11	5	10
	Russian	4	20	6	3
	Spanish	3	8	19	11
	French	2	11	10	22
Overall accuracy:		**38%**			

Table 4-11 4-gram in word level for set B

Set C (5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	23	12	3	7
	Russian	9	23	6	6
	Spanish	9	9	17	13
	French	10	12	9	16
Overall accuracy:		**40%**			

Table 4-12 4-gram in word level for set C

The performance of 4-gram algorithm in word level is low. But the accuracy rate is not the major problem of 4-gram algorithm. If we summarize all the numbers in each number, we'll get: 155 for set A, 160 for set B, and 180 for set C. Considered the total number of authors in each set is 200, a big number of them for each group are not able to be classified into any of the language groups.

Why is that? Through out checking the raw data, it is known that because the 4-word phrases have very low reappearance rate, therefore if the text for test is short, the pieces of 4-grams might not able to be found in any of the training set we've built. That's why a lot of them are not identified successfully.

And if we compare the results of unigram, bigram, trigram, and 4-gram algorithms in word level, it is obviously that bigram algorithm got the best result most of the 3 sets than all the other algorithms. It might indicate that the 2-word phrases are the most popular and the most representative combination of words.

4.6 Bigram in character level

Set A (0.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	26	5	9	10
	Russian	5	21	8	15
	Spanish	15	4	23	8
	French	15	5	17	13
Overall accuracy:		**42%**			

Table 4-13 Bigram in char level for set A

Set B (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	26	10	12	1
	Russian	5	33	11	1
	Spanish	7	3	36	3
	French	4	11	27	8
Overall accuracy:		**52%**			

Table 4-14 Bigram in char evel for set B

Set C (5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	21	12	15	2
	Russian	9	29	7	4
	Spanish	6	12	25	6
	French	9	8	25	8
Overall accuracy:		**42%**			

Table 4-15 Bigram in char level for set C

The performance is not quite good from the bigram in char level for most of the 3 sets. And it especially works badly for French.

4.7 Trigram in character level

Set A (0.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	22	3	7	18
	Russian	7	18	8	16
	Spanish	7	2	21	20
	French	9	4	9	28
Overall accuracy:		**45%**			

Table 4-16 Trigram in char level for set A

Set B (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	27	6	12	4
	Russian	1	33	9	7
	Spanish	2	3	37	7
	French	8	13	19	10
Overall accuracy:		**54%**			

Table 4-17 Trigram in char level for set B

Set C (5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	24	11	12	3
	Russian	2	31	14	2
	Spanish	3	10	32	4
	French	6	12	18	14
Overall accuracy:		**51%**			

Table 4-18 Trigram in char level for set C

We can see that the trigram algorithm in character level works better than bigram in general, especially better for Chinese, Russian, and Spanish, but not as good for French.

And it can be observed that the results for set B and set C are better than which of set A.

4.8 4-gram in character level

Set A (0.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	31	4	6	9
	Russian	5	16	13	15
	Spanish	7	2	24	17
	French	4	5	9	32
Overall accuracy:		**52%**			

Table 4-19 4-gram in char level for set A

Set B (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	31	5	9	4
	Russian	2	33	9	6
	Spanish	3	4	39	3
	French	8	12	17	13
Overall accuracy:		**58%**			

Table 4-20 4-gram in char level for set B

Set C (5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	24	9	14	3
	Russian	1	31	14	3
	Spanish	1	10	34	4
	French	6	7	18	19
Overall accuracy:		**54%**			

Table 4-21 4-gram in char level for set C

The 4-gram algorithms have the best performance among all the n-grams algorithms in character level. It's especially good for Chinese and Spanish, and also not bad for Russian and French.

It's indicates that the combination of 4 letters have the strongest representativeness in char level compared to 2 or 3 letters combination.

And the results for both set B and set C won over set A again, though quite close.

4.9 Spell check

Set A (0.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	17	2	3	3
	Russian	1	9	5	3
	Spanish	7	2	15	7
	French	10	4	5	16
Overall accuracy:		**29%**			

Table 4-22 spell check for set A

Set B (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	11	2	8	4
	Russian	7	8	7	8
	Spanish	3	7	25	5
	French	9	10	10	12
Overall accuracy:		**28%**			

Table 4-23 spell check for set B

Set C (5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	10	5	2	5
	Russian	3	9	8	2
	Spanish	2	2	21	4
	French	7	3	7	18
Overall accuracy:		**29%**			

Table 4-24 spell check for set C

The spell check met the same problem with the 4-gram algorithm in word level. A lot of authors are not able to be identified into any of the language groups. And we don't see much difference from the performance of the 3 training sets. It's understandable, because not everyone makes mistake when posting on social media.

And it might also indicate that a highly misspelt word is equally easy to be misspelt by people have different language background.

4.10 Test on the refined training set

From the data above, it can be found that:

1. In most conditions the set B and set C work better than set A, which means the performance of set A is limited by its size. Therefore, 0.5 million words for each language are not enough.
2. In the competition between set B of 1.5 million words and set C of 5 million words, set B wins in most of the conditions. Though we strengthened set C by size, but it didn't performed better because of the imported noises by lowering threshold selecting corpus.

From the summary above, we knew that the size of set B which is 1.5 million words is the most proper size among the 3 based on the resource we currently have. Thus we want to make a good training set in the size of 1.5 million words.

In the other hand, we do have observed that the 3 distinct training sets reflected some common feature for certain algorithm, which is just what we expected. Therefore we learned from the tests that the assumption made in Chapter 3.4 that authors with certain language background are more suitable for a set of some algorithms, but authors with another language background might be more suitable for a set of different algorithms. Given this feature, to boost the final result by combine the 8 algorithms in a weighted function are theoretically possible.

According the summary above, we refined the training set by applying more strict condition on selecting corpus, while limiting the size by 1.5 million words. For example, we limited people clearly coming from a certain country instead of possibly just ethnically belong to that country. And we tried to add some authors known to be good samples manually into the training set. Throughout doing that, we got a refined training set named as "set D". And here we are to test its performance.

As what is done in Chapter 4.2 to Chapter 4.9, we tested another 200 authors (50 authors for each language background with no duplicate from the training set). And the result is listed below:

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	38	6	5	0
	Russian	2	37	9	2
	Spanish	4	2	43	0
	French	7	4	20	19
Overall accuracy:		**69%**			

Table 4-25 Unigram in word level for set D

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	38	4	5	2
	Russian	1	40	5	3
	Spanish	1	2	43	3
	French	4	6	19	21
Overall accuracy:		**72%**			

Table 4-26 Bigram in word level for set D

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	34	4	4	7
	Russian	0	41	5	3
	Spanish	2	4	39	4
	French	3	8	17	21
Overall accuracy:		**68%**			

Table 4-27 Trigram in word level for set D

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	34	5	3	5
	Russian	3	37	3	2
	Spanish	3	7	33	5
	French	2	11	10	22
Overall accuracy:		**63%**			

Table 4-28 4-gram in word level for set D

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	27	9	13	0
	Russian	12	29	9	0
	Spanish	7	4	38	0
	French	15	10	22	3
Overall accuracy:		**49%**			

Table 4-29 Bigram in char level for set D

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	26	9	13	1
	Russian	6	32	11	1
	Spanish	2	2	44	1
	French	5	11	23	11
Overall accuracy:		**57%**			

Table 4-30 Trigram in char level for set D

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	37	3	8	1
	Russian	3	36	7	4
	Spanish	4	2	41	2
	French	5	7	19	19
Overall accuracy:		**67%**			

Table 4-31 4-gram in char level for set D

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	17	1	5	5
	Russian	5	24	8	0
	Spanish	4	2	35	0
	French	7	3	7	18
Overall accuracy:		**47%**			

Table 4-32 Spell check for set D

It's obvious to be observed that, the algorithm giving best result is still bigram in word level and 4-gram in character level algorithm, which means it obeys the same rule we observed before on set A, set B and set C. But the accuracy rate for each algorithm is clearly boosted in the meantime.

To obtain a good combine result above, there is a need to design a way to efficiently using the results of each algorithm. As it is discussed in Chapter 3.4, we developed a method weight an algorithm for a language background according its performance on that language.

We index the 4 language backgrounds as L_i (i=1~4), and index the 8 algorithms as A_j (j=1~8). And provided applying algorithm A_j, accuracy number of language L_i is N_{ji}, we assign the weighted value for Language L_i and A_j as W_{ji}:

$$Wji = \left(\frac{8 * Nji}{\sum_{j=1}^{8} Nji} \right)^5$$

By applying the weighted value on each algorithm, and voting on the result, we obtained the combined result in the confusion matrix below:

Set D (1.5 million)		Actual			
		Chinese	Russian	Spanish	French
Classified	Chinese	39	3	3	4
	Russian	1	40	6	3
	Spanish	3	1	41	4
	French	2	5	13	30
Overall accuracy:			**75%**		

Table 4-33 Combined Algorithm for Set D

4.11 Discussion

Therefore the overall accuracy rate of the combined algorithm is 75%. This is a good result proved our classifiers have an excellent identification rate. But if we compare the result comes from each algorithm, some interesting observations are to be shown.

First, the combined result is better than the result from any single algorithm. It proved our assumption that every algorithm has itselves advantages and limits. Each algorithm represents part of the influence from author's language background. For example, the bigram, trigram, and 4-gram algorithms in word level works very well for identifying Chinese authors. It might indicate the unique patterns of matching words and using phrases of authors who have Chinese background. The unigram, algorithm in word level, 4-gram algorithms in character level and the spell check algorithm work great for identifying Spanish authors. It might indicate the authors with Spanish background are more likely to make mistakes in spelling words.

Second, Chinese and Russian group have very good result for most of the algorithms, and the combined result does the same. But the French works not that good. A lot of authors with French background are judged to be Spanish. It's a quite reasonable phenomenon. Because we knew that the France and Spain are nearer to each other than to China or Russia in either aspect of geographic or language, so there should be more similarity between French and Spanish, which means they are possibly more difficult to be distinguished from each other.

Third, the n-gram algorithms in word level produced better result than the n-gram algorithms in character level and the spell checking algorithm in general. It might indicated that the influences from author's background are more likely to be reflected in the way he expresses, by using words or phrases, and less to be reflected by wrongly spelling words.

Fourth, the algorithms in word level work the best when the n is set to 2, it might reflect that people are more often to use 2-word phrases. And the algorithms in character level work the best when the n is set to 4, it might reflect people's spelling habits.

Overall, the observations we found in the test data logically fitted the common sense about the languages. It's a strong support of the cogency of our test result.

5 Conclusion and Future Work

5.1 Conclusion

In this thesis we have developed a method combined with n-gram algorithm in word level, n-gram algorithm in character level, and spell checking algorithm to identify the authors' native language from the English text they have written.

The combined result of classifying English text of authors with 4 native languages: Chinese, Russian, Spanish, and French has got a overall accuracy rate of 75% which is satisfying. And we have done tests on different sizes of training set to make the best balance between accuracy and efficiency. It would be definitely a powerful tool to determine a user's mother country independently. And it would also quite optimistic to work together with the Language indicator, Time-zone indicator, and Geo-terms indicators which are developed by Jaran Nilson and Ole-Alexander Moy, because the validating requirement is completely distinct from those algorithms mentioned above.

We are not aware of any previous work in the same area. The result of this research could be of significant value for those people studying the social media. And also it perhaps would be an inspiration for people working on text classification and data mining.

5.2 Future work

Limited by time, there are some ideas haven't been put into practice, but they might improve the result.

5.2.1 Expansion to other languages

As we have made the classifier on English for authors whose native language are Chinese, Russian, Spanish, and French. We may also try to expand the method to identify authors with other native languages. More than that, we might also targeting corpus in other language. For instance, it might even possible to classify Portuguese and Brazilian who both write in Portuguese. Those expansions might create even more exciting result.

5.2.2 Parts of speech analysis

It's is possible to extend the n-gram algorithms to the parts of speech analysis for given corpus. Through out the sequence of words with different parts of speech, it should indicate the grammar structure of a sentence. And of course we may assume that the grammar structure of a English speaker should relate to his/her native language.

The challenge of implementing parts of speech analysis mainly comes to the parts of speech tagging. There are some problems with the parts of speech tagging. First, a single word could have distinct parts of speech depending on the given context. Second, to construct a good dictionary with parts of speech tagged. It's hard to get parts of speech tagged dictionary with good coverage of words. And also the existence of great amount of derivative largely increased the difficulty of parts of speech tagging. But if we follow the rule of derivative words, it's still possible to find a way to tag them.

References

[1] Khirulnizam Abd Rahman (2011), Social Media Marketing [Online], Available: www.kuis.edu.my/fstm/ecommerce/ems/Social-Media-Marketing.ppt

[2] D. Steven White (2010), All Things Marketing [Online], Available: http://dstevenwhite.com/2010/08/08/social-media-growth-from-2006-to-2010/

[3] Laura Frangi (2010), We are social [Online], Available: http://wearesocial.net/blog/2010/08/uks-media-consumption-habits/

[4] Jaran Nilsen (2007), Locating discussion board users with Bayesian analysis of geographic terms, language and timestamps

[5] Ole-Alexander Moy (2008), Identifying Geographic Terms within Natural Language Text

[6] Dr Gordon Coates (2009), Notes On Communication, pp. 159-160

[7] R. O. Duda, P. E. Hart, and D. G. Stork (2001), Pattern Classification. Wiely-Interscience .

[8] Ioan Pop (2006), An approach of the Naive Bayes classifier for the document classification, General Mathematics Vol. 14, No. 4 (2006), 135–138

[9] William B. Cavnar and John M. Trenkle, N-Gram-Based Text Categorization

[10] Rishin Haldar and Debajyoti Mukhopadhyay(2011), Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach

[11] IBM (2004), Can't beat Jazzy, [Online] Available: http://www.ibm.com/developerworks/java/library/j-jazzy/

[12] Megan Bohensky, Technical Issues in Data Linkage [Online], Available: http://www.health.vic.gov.au/vdl/downloads/engagingwithpossibilities_presentations/technical_issues.pdf